SUB A2

## CLAIMS

**1.** An integrated circuit (IC) module comprising:

a processor;

volatile memory and nonvolatile memory operatively coupled to the processor; and

a file system to manage access to one or more data files stored in the volatile memory and in the nonvolatile memory.

**2.** An integrated circuit (IC) module as recited in claim 1, wherein the file system exposes a set of application program interfaces that are used by an application to request the one or more data files stored in the volatile memory and the nonvolatile memory.

**3.** An integrated circuit (IC) module as recited in claim 2, wherein individual functions defined in the set of application program interfaces include a parameter identifying whether an associated data file is stored in the volatile memory or the nonvolatile memory.

**4.** An integrated circuit (IC) module as recited in claim 1, wherein the file system comprises a memory region directory to identify whether the one or more data files are stored in the volatile memory or in the nonvolatile memory.

**5.** An integrated circuit (IC) module as recited in claim 1, wherein the file system comprises:

a memory region directory to identify whether the one or more data files are stored in the volatile memory or in the nonvolatile memory; and

one or more file location specifiers to specify a physical location of the one or more data files.

**6.** An integrated circuit (IC) module as recited in claim 5, wherein the file location specifier comprises a file allocation table.

**7.** An integrated circuit (IC) module as recited in claim 5, wherein the file location specifier comprises a file allocation table.

SUB A37 **8.** An integrated circuit (IC) module as recited in claim 1, further comprising at least one application stored in the nonvolatile memory and executable on the processor to request access to the one or more data files.

**9.** An integrated circuit (IC) module as recited in claim 1, embodied as a smart card.

**10.** An integrated circuit (IC) module comprising:

a processor;

volatile memory operatively coupled to the processor, the volatile memory storing volatile data in at least one data file;

nonvolatile memory operatively coupled to the processor, the nonvolatile memory storing nonvolatile data in at least one data file;

a memory region directory to identify whether a requested data file is located in the volatile memory or in the nonvolatile memory; and

a file location specifier to specify a physical location of the requested data file within the volatile memory or the nonvolatile memory identified by the memory region directory as containing the requested data file.

**11.** An integrated circuit (IC) module as recited in claim 10, wherein the data file stored in the volatile memory is destroyed when power is removed from the IC module.

**12.** An integrated circuit (IC) module as recited in claim 10, wherein the memory region directory is stored in the nonvolatile memory.

**13.** An integrated circuit (IC) module as recited in claim 10, wherein the file location specifier comprises:

a first file allocation table to specify physical locations of data files within the volatile memory; and

a second file allocation table to specify physical locations of data files within the nonvolatile memory;

**14.** An integrated circuit (IC) module as recited in claim 10, wherein:

the nonvolatile memory comprises both a read only memory and a read/write memory; and

the file location specifier comprises:

(a) a first table to specify physical locations of data files within the read only memory;

(b) a second table to specify physical locations of data files within the read/write memory; and

(c) a third table to specify physical locations of data files within the volatile memory.

**15.** An integrated circuit (IC) module as recited in claim 10, further comprising an application program interface to enable an application to access the data files in the volatile memory and the nonvolatile memory.

**16.** An integrated circuit (IC) module as recited in claim 10, further comprising an initialization mechanism to delete any data from the volatile memory and to remove any reference to data files in the volatile memory from the memory region directory.

**17.** An integrated circuit (IC) module as recited in claim 10, embodied as a smart card.

**18.** A file system for an integrated circuit module, comprising:

means for handling a request for a data file stored on the integrated circuit module;

means for identifying whether the data file is located in volatile memory or nonvolatile memory; and

means for specifying a physical location of the data file within the volatile memory or the nonvolatile memory.

**19.** A file system as recited in claim 18, further comprising means for deleting all data files in the volatile memory and for removing any reference to the data files that might be stored in the nonvolatile memory.

**20.** An operating system for an integrated circuit (IC) module, comprising:

a file system to manage access to data files stored in both volatile memory and nonvolatile memory; and

an application program interface (API) to expose the file system to applications.

**21.** An operating system as recited in claim 20 wherein the API defines a function for opening a data file, the function being used to open data files in the volatile memory and the nonvolatile memory.

**22.** An operating system as recited in claim 20 wherein the API defines a function, the function including a parameter specifying whether a data file resides in the volatile memory or in the nonvolatile memory.

**23.** An operating system as recited in claim 20 wherein the file system comprises:

a memory region directory to identify whether individual data files are located in the volatile memory or in the nonvolatile memory; and

a file location specifier to specify a physical location of the requested data file within the volatile memory or the nonvolatile memory.

**24.** An operating system as recited in claim 23 wherein the file location specifier comprises:

a first file allocation table to specify physical locations of data files within the volatile memory; and

a second file allocation table to specify physical locations of data files within the nonvolatile memory.

**25.** A file system for an integrated circuit module, comprising:

an application program interface to enable an application to access files stored in volatile memory and nonvolatile memory; and

a memory region directory to identify whether a file is stored in the volatile memory or the nonvolatile memory.

**26.** A file system as recited in claim 25, further comprising a file location specifier to specify a physical location of the file within the volatile memory or the nonvolatile memory.

**27.** A file system as recited in claim 25, further comprising:

a first file allocation table to specify physical locations of data files within the volatile memory; and

a second file allocation table to specify physical locations of data files within the nonvolatile memory.

**28.** A computer-readable medium storing computer-executable instructions that, when executed on a smart card, direct the smart card to:

store data in a volatile data file within volatile memory of the smart card; and

facilitate access to the volatile data file by one or more applications.

**29.** A method for operating an integrated circuit (IC) module, comprising:

receiving a request for a data file stored on the IC module;

identifying, within the IC module, whether the data file is located in volatile memory or nonvolatile memory; and

specifying a physical location of the data file within the volatile memory or the nonvolatile memory.

**30.** A method as recited in claim 29, further comprising exposing functions to manipulate the data files, the same functions being used regardless of whether the data files are located on the volatile memory or the nonvolatile memory.

**31.** A method as recited in claim 29, further comprising deleting all data files in the volatile memory and for removing any reference to the data files that might be stored in the nonvolatile memory.

**32.** A computer-readable medium storing computer-executable instructions that, when executed on a processor, perform the method as recited in claim 29.

**33.** A method comprising:

storing data in a volatile data file in volatile memory of an integrated circuit module;

receiving, from a requestor, a request to access the volatile data file on the integrated circuit module;

evaluating whether the requestor is authorized to access the volatile data file; and

in an event that the requestor is authorized, locating the volatile data file in the volatile memory.

**34.** A method as recited in claim 33, wherein the data stored in the volatile data file is produced by a first application and the requestor is a second application.

**35.** A method as recited in claim 33, wherein the locating comprises:

ascertaining that the volatile data file is located in the volatile memory; and

specifying a physical location of the volatile data file within the volatile memory.

**36.** A method as recited in claim 33, wherein the locating comprises passing in a parameter that identifies the volatile data file as being stored in the volatile memory.

**37.** A method as recited in claim 33, further comprising returning a handle to the volatile data file.

**38.** A computer-readable medium storing computer-executable instructions that, when executed on a processor, perform the method as recited in claim 33.

**39.** A method comprising:

storing volatile data in at least one volatile data file in volatile memory;

storing nonvolatile data in at least one nonvolatile data file in nonvolatile memory;

receiving a request to access a particular data file;

determining whether the particular data file is stored in the volatile memory or the nonvolatile memory; and

locating the particular data file.

**40.** A method as recited in claim 39, wherein the locating comprises using a file allocation table to locate the particular data file.

**41.** A method as recited in claim 39, further comprising exposing a common set of functions to manipulate both the volatile data files and the nonvolatile data files.

**42.** A computer-readable medium storing computer-executable instructions that, when executed on a processor, perform the method as recited in claim 39.

**43.** A method comprising:

storing data produced by a first application within a volatile data file within volatile memory in a smart card; and

accessing the volatile data file from a second application.

**44.** A method as recited in claim 43, further comprising evaluating whether the second application is authorized to access the volatile data file.

**45.** A computer-readable medium storing computer-executable instructions that, when executed on a processor, perform the method as recited in claim 43.

ADD A47